Triple

TRANSFORMING RESEARCH THROUGH INNOVATIVE PRACTISES FOR LINKED INTERDISCIPLINARITY EXPLORATION

Project Number: 863420 Start Date of Project: 01/10/2019 Duration: 42 months

Architecture Decision Record

SCRE_009 - Authors management

last modified: 19/04/2022

Status

Proposed

Context

SCRE must also take care of the management of the "People" index of the GoTriple Elasticsearch index. This index is populated from the authors field of publications, a process which causes plenty of problems including:

- recognizing "real persons". Sometimes you find in the authors things like "Department of Computer Science" or "ACM Conference 2021"
- identifying alternative spelling for the same person, e.g. Suzanne Dumouchel, Dumouchel, Suzanne, Dumouchel, S., S. Dumouchel
- identifying homonyms.

Decision

The expert at the last TRIPLE review suggested to use for this problem an authority registry, e.g. VIAF. At a first glance VIAF contains a lot of noise and wouldn't be helpful for our approach. See for example these searches:

- http://viaf.org/viaf/search?query=local.personalNames%20all%20%22mounier%20pierre%22&sortKeys=holdingscount&recordSchema=BriefVIAF
- http://viaf.org/viaf/search?query=local.personalNames%20all%20%22francesca%20di%20donato%22&sortKeys=holdingscount&recordSchema=BriefVIAF
- http://viaf.org/viaf/search?query=local.personalNames%20all%20%22di%20donato%2C%20F.%22&sortKeys=holdingscount&recordSchema=BriefVIAF

Also to speed up processing it would be advisable to have a solid rule-based approach that could allow us to reduce at the minimum the noise, ensuring a decent precision at the risk of reducing the number of profiles shown. In short, better less but good!

What follows is a possible matching algorithm inspired by these articles, which has been adapted to the metadata that we have at hand:

- https://arxiv.org/abs/1308.0749
- https://arxiv.org/pdf/2103.14558.pdf

We assume to have an **Authors cache managed by SCRE** with:

- author full name (null if not available: see below)

TRIPLE has received funding from the EU Horizon 2020 Research and Innovation Programme (2014-2020); H2020-INFRAEOSC-02-2019, under the agreement No.863420

- combinations (see below)
- initials (see below)
- keywords and their weight
- min year of publication
- max year of publication
- links to the recognised publications/duplicates clusters
- potential_duplicate flag.

If the latter is TRUE the author is not persisted in the GoTriple index.

Given a publication we take:

A. the authors
B. the original keywords
C. the publication year
D. the publication ID *and* its possible duplicate cluster ID

Then:

1. we transform from Unicode to ASCII the author string (by using JUnidecode)
2. if the author is
    a. longer than N characters (e.g. 30) or shorter than L characters (e.g. 8)
    b. AND it doesn't contain a space
    c. => discard
3. if the author doesn't contain a comma
    a. => we assume it's a full name (e.g. Ana Paula Veloso de Linhares)
    b. => we extract the possible combinations and initials
        i. Linhares, Ana Paula Veloso de -> Linhares, A.
        ii. de Linhares, Ana Paula Veloso -> de Linhares, A.
        iii. Veloso de Linhares, Ana Paula -> Veloso de Linhares, A.
        iv. Paula Veloso de Linhares, Ana -> Paula Veloso de Linhares, A.
4. if the author contains a comma we consider it a "combination"; we take it plus the variant with the first initial
    a. Veloso de Linhares, Ana Paula -> Veloso de Linhares, A.
5. if the author contains a comma followed by 1 character and a "." we consider it an initial. Otherwise if there are more characters followed by a "." we transform it as an initial.
    a. Veloso de Linhares, A.

We search in the Authors cache for all authors whose potential_duplicate flag is FALSE, giving a score for each possible match.

1. if we don't have a match (first time author) we create the entry in the authors cache with
    a. full name (or null), the variants (or null), the initials
    b. the current publication's keywords
    c. the current publication's ID
    d. (if any) the current publication's cluster ID
    e. the year of the present publication (min = max = year).
    f. The procedure stops.
2. we have a match. We assign a score (**NOTE: numbers are just examples**):
    a. 10 for full name
    b. 8 for a match with a combination
    c. 5 for a match with an initial IF one the full names isn't available
        i. this to avoid creating a match for Di Donato, F. which can correspond either to Di Donato, Francesca AND Di Donato, Flora.
3. we check the keywords of the publication
    a. 1 point for the each match * the keyword weight
4. check the year of publication
    a. if it is included in the min-max range of the author => 10
    b. if it's 3 years below or over the range => 8
    c. if it's 5 years below or over the range => 5
    d. if it's 10 years below or over the range => 1

5. if the current document is in the cluster of the author's document => 20
6. we sum the score and if it's >= 15 we assume it's the same author.
    a. we add the publication ID and its eventual cluster ID in the authors cache
    b. we merge the keywords by recalculating their weight. If a keyword matches => its weight++
    c. if needed, we update the min-max publication years range
    d. we assign the publication to the matched author
7. if the score < 15
    a. ~~either we discard the author (pro: less noise; cons: future matches aren't possible)~~
    b. ~~or we consider it as a separate author (pro: in the future there might be a match with it; cons: we create potential duplicates).~~
    c. we store it as "potential duplicate" in the cache. We don't create an entry in GoTriple out of this data but if a new entry doesn't match, it can be searched with the steps described above on all potential duplicates. If there's a match, the author is not a potential duplicate anymore (potential_duplicate = NULL/False).

To increase the precision we also might decide to export on the GoTriple Index **ONLY** the authors with >= 2 publications.

Consequences

SOFTWARE DESIGN MUST GO HERE.